Recent work in multiphase flow at NETL

Madhava Syamlal
Focus Area Leader
Computational and Basic Sciences
madhava.syamlal@netl.doe.gov

presented at
BHEL Workshop on Modeling Coal
Combustion and Gasification
Tiruchy, India
October 29- November 1, 2007

National Energy Technology Laboratory
Morgantown, WV
Outline of Presentation

- MFIX-NG
- Polydispersed Flow
- High Resolution Gasifier
- Future Outlook
MFIX-NG – Objectives

1. Improve the fidelity of multiphase flow models
   • Balance equations and constitutive relations
   • Validation studies

2. Develop fast and accurate numerical techniques
   • Accurate spatial and temporal discretization
   • Scalable parallel algorithms
   • Verification studies

3. Develop advanced post-processing capabilities
   • Data analysis
   • Visualization
   • Reduced order models

Need a software platform for these
The Path Forward

• Continue using/developing MFIX
  • Needs much programming
  • Cannot easily reuse modern software components
  • Cannot take advantage of available open source software written in modern programming languages

• Use commercial software
  • No flexibility for the concurrent development of theory, numerics, validation, and application
  • May need to abandon MFIX open source users

• Construct software from existing software components
  • Achieve desired features in the software
  • Reduced development cost
MFIX-NG Primary (Programmatic) Goals

- Develop software infrastructure to model multiphase flow processes in power and process industry (e.g., coal gasifiers).
- Develop and validate multiphase flow theory:
  - Transport equations,
  - Boundary conditions, and
  - Constitutive relations.
- Develop numerical techniques for solving these multiphase flow equations efficiently and accurately.
- Increase the use of modeling for design and control in power and process industry.
MFIX-NG Secondary (User-Related) Goals

- Enable scientists to focus on model and algorithm development and validation, rather than code development and debugging.
- Reduce the need for scientist to understand the details of the underlying software framework, e.g., parallel computing.
- Reduce the development time for new applications by leveraging existing software and solver technology.
- Allow computational scientists to explore new algorithms through the use and modification of existing software.
Unique Features Sought in MFIX-NG

- **Script-based front-end**: Physical models and numerical techniques are expressed in a thin layer of top-level code
  - Fast, error-free development of novel physical models and numerical techniques.
- **Components-based design**: Software system is composed of replaceable units of code with well-defined interfaces
  - Reuse advanced software components developed at other national labs and universities
- **Open-source development**: Process and infrastructure to validate and accumulate code contributions from users
  - Platform for exchanging and validating ideas
  - User contributions deepen software capabilities
Open Source Development

- Open Source (OS) Software
  - Term coined in 1998
  - Source code freely available for study, change and reuse
  - Comes with a license (GPL, LGPL, …) that requires users to follow certain rules regarding usage and redistribution
  - Success well established: Linux (operating system), Apache (Webserver)

- Advantages
  - Verification by ‘many eyes’
  - Accumulation of user contributions
  - Easier to exploit super computers
  - Information dissemination
  - Better peer review
Goals, Objectives and Challenges (I)

- Building on the success of the MFIX flow solver, the next-generation solver will offer
  - Greater geometric and modeling flexibility
  - Ability to model risers with inlets, cylinders without a centerline in grid, even entire apparatus
  - Easier extendibility and maintainability.

- New solver will initially re-implement the MFIX solution algorithms and physical models.

- It will leverage externally developed libraries to easily take advantage of improvements made by other researchers.

- At the conclusion of the project, the new solver will
  - Mimic the modeling capabilities now available in MFIX
  - Offer greater geometric flexibility by the use of unstructured meshes
  - Provide a platform for advanced multiphase solution algorithm development
MFIX-NG Development – 1

- Following the Unified Modeling Process adapted to the unique needs of the current project (e.g., application domain, research code, legacy codes, collaboration).
  - User Requirements Document (URD)
  - Software Requirements Document (SRD)
  - Software Design Document (SDD)
  - Software Development Plan (SDP)
  - Iterative Development process: Development goals are set every quarter based on a two-year roadmap.
  - Software Development and Testing
MFIX-NG Development –3

- Software Requirements Document (SRD)
  - Explored and qualitatively evaluated a set of alternative frameworks and PSEs based on the available documentation, examples, etc.
  - Rated each package with respect to features listed in SRD
  - Conducted user survey to determine the relative importance of various features listed in the SRD
  - Compiled the results of user survey and package ratings for determining the optimal framework
MFIX-NG Development – 4

- Software Design Document (SDD)
  - Use cases (typical ways in which the software will be used)
  - class design diagram, collaboration diagrams, ...External interface requirements (user, software, hardware).
- Performance targets, quality assurance.
- Design constraints.
- Preliminary design - top-level architecture, main components, dataflow charts.
- Testing strategy: unit tests, integration tests, verification tests
MFIX-NG Development –5

- Software Development Plan (SDP)
  - Tasks/features, benefits (limitations)
  - Completion criteria
  - Schedule/milestones
  - Resources

- Software Development and Testing
  - Two-year development time to achieve full capabilities
  - Scheduled to finish FY08-Q3
  - Approximately 2 FTE/year development effort

- MFIX will be maintained as a well-tested legacy code
OpenFOAM Selected

- Selection of the basic software libraries was the first milestone.
- Exhaustive evaluation of linear solvers, problem-solving environments, and CFD-specific libraries was performed.
- The OpenFOAM library was chosen as the most suitable.
## Framework Selection

<table>
<thead>
<tr>
<th>Framework</th>
<th>Organization</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Weighted Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenFOAM</td>
<td>OpenCFD, UK</td>
<td>46.1</td>
<td>52.0</td>
<td>39.6</td>
<td>100.0</td>
<td>58.8</td>
</tr>
<tr>
<td>Trilinos</td>
<td>CCM, Sandia</td>
<td>3.9</td>
<td>26.0</td>
<td>56.3</td>
<td>100.0</td>
<td>41.1</td>
</tr>
<tr>
<td>SAMRAI</td>
<td>CASC, Livermore</td>
<td>0.0</td>
<td>34.0</td>
<td>41.7</td>
<td>62.5</td>
<td>30.6</td>
</tr>
<tr>
<td>OVERTURE</td>
<td>CASC, Livermore</td>
<td>3.9</td>
<td>38.0</td>
<td>0.0</td>
<td>50.0</td>
<td>22.5</td>
</tr>
<tr>
<td>PETSc</td>
<td>Argonne</td>
<td>23.1</td>
<td>0.0</td>
<td>0.0</td>
<td>50.0</td>
<td>19.2</td>
</tr>
<tr>
<td>AMROC</td>
<td>CACR, CalTech</td>
<td>0.0</td>
<td>20.0</td>
<td>0.2</td>
<td>75.0</td>
<td>22.4</td>
</tr>
<tr>
<td>ROCCOM</td>
<td>U. Illinois</td>
<td>7.7</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td>WEIGHTS (Avg. of users)</td>
<td>0.335</td>
<td>0.255</td>
<td>0.18</td>
<td>0.230</td>
<td></td>
</tr>
</tbody>
</table>

Objectives: 1 – Physics representation; 2 – Numerical scheme; 3 – SW development/maintenance; 4 – Open source

Gel et al. (2006)
OpenFOAM Strengths

- Long development history. Marketed as a commercial product for many years.
- Code base was open sourced. A large active user group grew up quickly. Many contributors from both academia and industry. As improvements are made to the OpenFOAM core libraries, NETL’s next-generation solver will be able to make use of them.
- OpenFOAM is object oriented and highly CFD specific, allowing for concise expression of CFD algorithms. Transport equations are expressed in terms of tensor calculus (i.e., \( \text{div}(\mathbf{U}), \text{grad}(p) \)) rather than as coefficients stored in indexed matrices and vectors.
- OpenFOAM natively supports unstructured meshes, high-order discretization schemes, parallel processing without need for special coding practices.
- OpenFOAM framework exists for treating Lagrangian particles. Can be used for DEM or DPM models.
- Also supports moving meshes. Can model moving baffles, etc.
Highlights for OpenFOAM

- Standalone CFD code (OOP, templated C++)
- Originated from Imperial College (1993)
- FOAM was offered as a commercial code
- Released as open source in December 2004
- 3rd party software dependencies: MICO, Paraview, gcc, LAM/MPI
- Active community with users group lists
OpenFOAM Example Code

- Simple example showing the creation of a transport equation for single-phase momentum
- The object $U$ is a vector field object representing the velocity in every cell in the domain. The object $UEqn$ is a vector matrix that represents the linear system defining the velocity field.
- The transient, convection, diffusion, and pressure terms are all neatly defined using the OpenFOAM notation.
- $\text{solve}(UEqn)$; solves the linear system for the new velocity field.

```
fvVectorMatrix UEqn
  (  
    fvm::ddt(U)  
    + fvm::div(phi, U)  
    - fvm::laplacian(nu, U)  
    == -fvc::grad(p)  
  );
  solve(UEqn);
```

$$
\frac{\partial U}{\partial t} + \nabla \cdot (UU) - \nabla \cdot v \nabla U = -\nabla p
$$
MFOAM – 1

- **Initial efforts focused on learning the OpenFOAM library and building a prototype two-phase code.**
  - Followed the prototype single-phase solver codes included with the OpenFOAM source code.
  - Included isothermal gas-solids transport.
    - Included partial elimination algorithm for drag terms to promote solution stability.
    - Direct solution of the solids volume fraction equation.
    - Hard-wired solids pressure relation and constant viscosities.
  - Prototype mFoam code was successful in baseline testcases and served as the basis for the expanded solver.
  - Simple ozone decomposition chemistry and oxygen and ozone transport equations were added to replicate a well-studied MFIX case. Used as a proof-of-concept solver for reacting flows.

- **Developed a post processing program to serve as a prototype to replicate PostMFIX capabilities.**
MFOAM – 2

- Using the mFoam prototype code, several researchers began implementation of the core MFIX model functionality.
  - Made use of CVS to maintain consistency during development process.
- A new storage scheme was devised to generalize the number of phases.
  - The n-phase partial elimination algorithm was implemented for the momentum equations. mFoam could then solve flows involving a single continuous phase and one or more dispersed phases.
- The gas transport equation was modified to include the compressibility term allowing for variable density gas flows.
- The energy transport equation was added to each phase.
  - The n-phase partial elimination algorithm was used to improve inter-phase coupling due to heat transfer.
MFOAM – 3

- A general framework was assembled for handling drag models. All of the MFIX drag models were transferred to mFoam using this new framework.
- The MFIX algebraic and PDE kinetic theory models were implemented, including the Johnson and Jackson BCs.
- The Schaeffer and Princeton frictional models
- The Ahmadi and Simonin turbulence models
- The momentum equations have been modified to include an additional term that allows for the solution of periodic flows driven by a pressure gradient.
- A baseline automated test harness has been set up to automatically compile and test new code revisions and flag any deviation from prior results.
• Adaptive time stepping algorithm used in MFIx was migrated to mFoam.
  – This increases both the computational efficiency and stability by increasing the timestep when the solution converges smoothly and by recovering from failed time steps.

• Modifications made to the momentum equations to remove spuriously large solids velocities when volume fraction is low.
  – Mirroring the approach in MFIx, this modification prevents the momentum equation from becoming poorly conditioned in very dilute flow regions.
MFOAM – 5

Unstructured Mesh – no centerline.

Spouted Bed Simulation
MFOAM – 6

- Evaluated four formats for reaction schemes and selected Cti format
  - MFIIX: easy to convert existing MFIIX reaction files; non-standard format
  - Chemkin: widely-used format; OpenFOAM already has a reader; format not extensible to multiple phases; fixed-column format prone to errors.
  - OpenFOAM: similar to Chemkin format. the data is easier to input than Chemkin; format would be limited to the use with MFOAM/OpenFOAM.
  - Cti: used in Cantera; easy to write and read; utilities for converting Chemkin files into cti format; enables integration with Cantera; will need a translator
Block Solver Development

Block Solver Development

• Equation Segregation in OpenFOAM: Matrix Support.
  – Linear system and solver classes in OpenFOAM currently support scalar coefficients only.
  – As matrix coefficients are scalar, equation segregation is enforced: for coupled systems or vector and tensor variables, each component is solved in turn.
  – Segregated solvers do not provide sufficient level of coupling: a block matrix and solver approach is needed.

• Handling Complex Coupling
  – For coupled vector and tensor variables, basic sparseness pattern follows from mesh connectivity. For efficient solution, this fact should be used
  – Two types of coupling
    • Inter-variable coupling: vector or tensor components coupled to each other.
    • General matrix-to-matrix coupling: multiple transport equations solved together, with implicit handling of linearized coupling terms.
  – Both approaches produce the same basic effect: choice will be made based on convenience in discretization and matrix assembly.
Sparse Matrix with Block Coefficients

- For cases of coupled vector and tensor variables, the FVM sparseness pattern is preserved: a vector component is coupled to other vector components in the same cell or to vector components in a neighboring cells.

- Example: block-coupled vector equation
  - Variable organization: \((u_x, u_y, u_z)\).
  - Ordering of each list matches the cell ordering and sparseness pattern matches the mesh.

\[
A_P = \begin{bmatrix}
[u_x \leftrightarrow u_x] & [u_x \leftrightarrow u_y] & [u_x \leftrightarrow u_z] \\
[u_y \leftrightarrow u_x] & [u_y \leftrightarrow u_y] & [u_y \leftrightarrow u_z] \\
[u_z \leftrightarrow u_x] & [u_z \leftrightarrow u_y] & [u_z \leftrightarrow u_z]
\end{bmatrix}
\]

\[
A_P u_P + \sum_N A_N u_N = R
\]

- \(A_P\) and \(A_N\) coefficients are tensors; the rest of linear algebra generalizes naturally, including vector-matrix multiplication and linear solvers.
MFOAM Future Work – 2

General Matrix-to-Matrix Coupling

- In cases where multiple equations for multiple variables are coupled in a general manner, a block coefficient approach is not appropriate: requires re-ordering of coefficients.
- Each matrix is assembled in isolation and placed into a block system. Example u1 – u2 coupling.

\[
\begin{bmatrix}
[u_1] & [u_1 \rightarrow u_2] \\
[u_2 \rightarrow u_1] & [u_2]
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2
\end{bmatrix}
= \begin{bmatrix}
R_1 \\
R_2
\end{bmatrix}
\]

- \([u_1]\): sparse matrix block containing \(u_1\) equation with implicit coupling.
- \([u_1 \rightarrow u_2]\): off-diagonal block matrix containing \(u_1 \rightarrow u_2\) coupling terms.
- Linear solver operates on a complete block system; preconditioning is performed on diagonal blocks only.
- Implementation involves multiple blocks and arbitrary coupling: \([u_1 \rightarrow u_2]\) is considered a coupled interface on the \([u_1]\) block.
- Linear algebra and linear solver algorithms now operate on a block system: \([u_1 \rightarrow u_2]\) is considered a single variable.
MFOAM Future Work – 3

- **Validation and Verification of mFoam**
  - During the current fiscal year, a testing protocol will be established.
    - Will compare mFoam results to MFIX and other CFD codes as a means of verification.
    - Will compare mFoam results to available experimental data as a means of validation.
  - These tests will be carried out in the following fiscal year.
  - Also in FY08, a careful investigation of high-order differencing schemes will be performed to mirror similar work done using MFIX by Guenther and Syamlal (2003).

- **Performance evaluation**
  - A comprehensive test will be made of the computational efficiency and scalability of mFoam. This will address single CPU performance as well as parallel performance over a tens to hundreds of CPUs.

- **Advanced algorithm development**
  - After replicating the capabilities, new solution algorithm approaches will be investigated. This will be the primary focus of research going forward.
  - These will make use of the block solver capabilities as they become available.
  - Also use of fractional step and explicit solution approaches.
  - Investigate multiphase flux limiting schemes that use the drag to compute and cap fluxes for all phases in a nonlinearly coupled manner.
Outline of Presentation

- MFIX-NG
- Polydispersed Flow
- High Resolution Gasifier
- Future Outlook
Polydispersity – Goals

- Implementation of I-A theory in MFIX (www.mfix.org)

- Verification and validation of theory with:
  - Experimental data produced at NETL, U. of Colorado and PSRI as well as data from the literature.
  - Numerical-experiments using discrete particle techniques.

- Application of theory to large-scale coal gasifier
Motivations and Justification

- Powders found in nature and those used in industry usually have wide size distribution. E.g. coal used in transport gasifiers
Motivation: results of simple shear flow

Energy dissipation in IA theory: \( N_{ip} = N_{s,ip} (T_p - T_i) + N_{sT,ip} + N_{svp,ip} \nabla \cdot \mathbf{v}_p + N_{svi,ip} \nabla \cdot \mathbf{v}_i \)

\( \frac{\rho_1}{\rho_2} = 1, \ \varepsilon_s = 0.3, \ \frac{\varepsilon_{s1}}{\varepsilon_{s2}} = 0.5, \ e = 0.95 \)

\( \frac{\rho_1}{\rho_2} = 1, \ \varepsilon_s = 0.3, \ \frac{\varepsilon_{s1}}{\varepsilon_{s2}} = 0.5, \ e = 0.8 \)

Brief description of I-A theory

\[
\begin{align*}
\frac{\partial}{\partial t} (\varepsilon_i \rho_i v_i) + \nabla \cdot (\varepsilon_i \rho_i v_i v_i) &= -\nabla \cdot (P_{ki} + P_{ci}) + \frac{\varepsilon_i \rho_i}{m_i} F_{iext} + \sum_{p=1}^{N} F_{Dip} \\
\sum_{i p p}^{Dip} \rho_i &+ P_{ci} = P_{s,i} I - 2 \mu_i^1 S_i - \xi_i^1 tr(D_i) I - 2 \sum_{p=1}^{N} (\mu_{ip}^2 S_p) - \sum_{p=1}^{N} (\xi_{ip}^2 tr(D_p) I) \\
F_{Dip} &= F_{s,ip} (v_p - v_i) + F_{snu,ip} (n_p \nabla n_i - n_i \nabla n_p) + F_{sTi,ip} \nabla T_i + F_{sTp,ip} \nabla T_p \\
\frac{3 \varepsilon_i \rho_i D \Theta_i}{2 m_i} + \nabla \cdot (q_{ki} + q_{ci}) + (P_{ki} + P_{ci}) : \nabla v_i &= \sum_{p=1}^{N} (N_{ip} - v_i \nabla F_{Dip}) \\
q_{ki} + q_{ci} &= -K_{sTi,i} \nabla \Theta_i + \sum_{p=1}^{N} K_{sTp,ip} \nabla \Theta_p \\
+ \sum_{p=1}^{N} K_{sv,ip} (v_i - v_p) + \sum_{p=1}^{N} K_{snu,ip} (n_i \nabla n_p - n_p \nabla n_i) \\
N_{ip} &= N_{s,ip} (\Theta_p - \Theta_i) + N_{sTi,ip} + N_{snu,ip} \nabla \cdot v_p + N_{svi,ip} \nabla \cdot v_i
\end{align*}
\]
Accomplishments

- I-A theory was implemented in MFIX open-source CFD code.

- I-A model modifications include:
  - Removed drag term contribution to granular energy equation.
  - Granular stresses and fluxes are now additive.
  - Correct dilute limit for granular viscosity and conductivity.
  - Modified Johnson-Jackson and Jenkins wall BC’s.

- Other code enhancements for polydisperse systems:
  - Added 3 radial distribution functions.
  - Added gas/solids drag law based on LBM.

- Code verified for simple granular shear flow.

- Code validation with experimental data:
  - Currently under way using Joseph et al. (2007) data.
  - Qualitative validation in dilute riser flow.
Assessment of Kinetic Theory in Dense Fluidized Beds

• Assess the validity of the binary kinetic theory of Iddir & Arastaoopour theory (IA) (2005)
  – Compare with axial segregation results of Joseph et al. (2007): experiments of a bidisperse fluidized bed at low gas velocities

• Assess the impact of a kinetic theory model in bidisperse fluidized beds and identify dominate terms:
  – Compare predictions from:
    • rigorous binary kinetic theory model of IA
    • formally monodisperse kinetic theory model of Lun et al. (1984)
    • without any kinetic theory model

Joseph et al. (2007)
Assessment of Kinetic Theory in dilute riser

- Assess the validity of the binary kinetic theory of Iddir & Arastaopour theory (IA) (2005)
  - Compare with radial segregation results of Mathiesen et al. (1999): experiments of a bidisperse dilute riser flow

- Assess the impact of a kinetic theory model in bidisperse riser flow and identify dominate terms that cause lateral segregation
Fully-developed upward dense gas/solids flow in a channel

1-D channel has 1 computational cell in flow-wise direction.

2-D channel has cell aspect ratio equal to one.

Johnson-Jackson wall BC with $\phi = 1e-04$ and $e_w = 0.7$.

Upward flow with periodic boundaries

10 cm width

Glass beads: 2.4 g/cm³ density
200 & 120 micron diameter
0.95 restitution coef.

Gas sup. Velocity fixed at: 5.5 m/s
Avg Solids volume fraction fixed at: 3%

Three concentration of initial powder mixtures was used (10, 50 and 90%).
Time-averaged solids volume fraction profiles

![Graph](image)

- dp = 200 microns
- dp = 120 microns
- 1-epg

**X [cm]**

**\( \varepsilon_s \)**

**a- Volume fraction profiles**
Time-averaged lateral momentum of major source terms

**a- time-average source terms in U-mom for phase 1 (large)**

- \( \nabla P_s \)
- \( F_{sn.i.p} (n_p \nabla n_i - n_i \nabla n_p) \)
- \( F_{sT.p} \nabla T_i + F_{sT.p} \nabla T_p \)
- \( F_{s.p} (v_p - v_i) \)

**b- time-average source terms in U-mom for phase 2 (small)**

- \( \nabla P_s \)
- \( F_{sn.i.p} (n_p \nabla n_i - n_i \nabla n_p) \)
- \( F_{sT.i.p} \nabla T_i + F_{sT.p} \nabla T_p \)
- \( F_{s.p} (v_p - v_i) \)
Time-averaged solids volume fraction profiles without S-S drag terms

a- Volume fraction profiles

b- Granular temperature profiles
Time-averaged particle diameter (number averaged) profiles using the full I-A theory

From experimental data of Mathiesen et al. (Int. J. of multiphase flow, 26 (2000) 387-419) using 120 and 185 microns glass beads.
Polydispersity Summary

• I-A theory was implemented in MFIX and verified using a simple granular shear flow.

• I-A theory comparison with experiments:
  – predicts the right segregation trends in dilute riser flow.
  – Disagreements observed in dense systems are currently under investigation\(^1\).

Outline of Presentation

- MFIX-NG
- Polydispersed Flow
- High Resolution Gasifier
- Future Outlook
High Res Gasifier Objectives

- **Programmatic goal:** High resolution (~1 mm grid-size) gasifier simulations to help with the design of commercial-scale gasifiers

- **Project goal:** Capability to sufficiently resolve the CCPI gasifier
  - Conduct a 10 M grid production simulation of CCPI gasifier with roughly one week turn-around time
  - Develop the capability for conducting such MFIX simulations
Challenges

• Simply running the gasifier simulation on a large number of processors (cores) will not help
  – Parallel efficiency reduces below 50,000 cells per processor in typical CFD codes → at most 200 cores, ~ ten weeks run time
• Need to use 2000 cores to reduce the turn around time
  – Current gasifier simulations use 10’s of cores and achieve 20 GFlop/s
  – Increase the number of cores by 100x and the speed to 1 TFlop/s
• Current mode of I/O handling introduces a bottleneck when the number of cores is increased
  – e.g., In a 2M-cell case the I/O time increased from 1.4 % on 16 cores to 5.9 % on 64 cores
• Two things must be done to enable high resolution simulations
  1. Increase the parallel scaling efficiency of MFIX
  2. Develop the ability to do distributed I/O from 100 to 1000’s of cores
Benchmarking

- Setup standardized test cases of increasing complexity (A. hydrodynamics only, B. simple chemistry, C. char combustion, D. coal gasification and combustion)
- Benchmarking for coarse (262 K cells) and fine grid (2 million cells) performed for Cases A & B.
  - Cray XT3 (PSC), AMD Opteron cluster (NERSC), and Cray XT4 (NCCS)
- Extensive TAU-based profiling of MFIIX showed the need for reducing global collective operations
Improve Existing Algorithm

- Reduced the number of global collective operations (dot products) in BiCGStab linear equation solver from 6 to ~3
  - Reduced the communication cost by 50%
- The net speed increase was found to be 10%
Global collectives become expensive on a large number of cores because of high latency

- Communication time = Latency + (message size)/(band width)
- Latency is the time needed to initiate a message transfer and bandwidth is the rate of message transfer
- The size of messages passed during collective operations on a large number of cores is small
- Then latency accounts for most of the communication cost
Standard BiCGStab iteration needs 6 vector-vector dot products (Original Algorithm)

Compute \( r^{(0)} = b - A x^{(0)} \) for some initial guess \( x^{(0)} \)

Choose \( \tilde{r} \) (for example, \( \tilde{r} = r^{(0)} \))

\[
\text{for } i = 1, 2, \ldots \\
\quad \rho_{i-1} = \tilde{r}^T \tilde{r}^{(i-1)} \\
\quad \text{if } \rho_{i-1} = 0 \text{ method fails} \\
\quad \text{if } i = 1 \\
\quad \quad p^{(i)} = r^{(i-1)} \\
\quad \text{else} \\
\quad \quad \beta_{i-1} = \left( \rho_{i-1}/\rho_{i-2} \right) \left( \alpha_{i-1}/\omega_{i-1} \right) \\
\quad \quad p^{(i)} = r^{(i-1)} + \beta_{i-1} (p^{(i-1)} - \omega_{i-1} v^{(i-1)}) \\
\quad \text{endif} \\
\quad \text{solve } M \hat{p} = p^{(i)} \\
\quad v^{(i)} = A \hat{p} \\
\quad \alpha_i = \rho_{i-1}/\tilde{r}^T v^{(i)} \\
\quad s = r^{(i-1)} - \alpha_i v^{(i)} \\
\quad \text{check norm of } s; \text{ if small enough: set } x^{(i)} = x^{(i-1)} + \alpha_i \hat{p} \text{ and stop} \\
\quad \text{solve } M \hat{s} = s \\
\quad t = A \hat{s} \\
\quad \omega_i = t^T s, t^T t \\
\quad x^{(i)} = x^{(i-1)} + \alpha_i \hat{p} + \omega_i \hat{s} \\
\quad r^{(i)} = s - \omega_i t \\
\quad \text{check convergence; continue if necessary for continuation it is necessary that } \omega_i \neq 0 
\]

end
Eliminated 2 dot products and combined two into one operation (New Algorithm)

Compute $r^{(0)} = b - Ax^{(0)}$ for some initial guess $x^{(0)}$
Choose $\tilde{r}$ (for example, $\tilde{r} = r^{(0)}$)

for $i = 1, 2, \ldots$

1. $\rho_{i-1} = \tilde{r}^T r^{(i-1)}$
   if $\rho_{i-1} = 0$ method fails
   if $i = 1$
     $p^{(i)} = r^{(i-1)}$
   else
     $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})(\alpha_{i-1}/\omega_{i-1})$
     $p^{(i)} = r^{(i-1)} + \beta_{i-1}(p^{(i-1)} - \omega_{i-1} v^{(i-1)})$
   endif
   solve $M \hat{p} = p^{(i)}$
   $v^{(i)} = A \hat{p}$
   $\alpha_i = \rho_{i-1}/\rho_{i}$
   $s = r^{(i-1)} - \alpha_i v^{(i)}$
   check norm of $s$, if small enough: set $x^{(i)} = x^{(i-1)} + \alpha_i \hat{p}$ and stop
   solve $M \hat{s} = s$
   $t = A \hat{s}$
   $\omega_i = t^T s$
   $x^{(i)} = x^{(i-1)} + \alpha_i \hat{p} + \omega_i \hat{s}$
   $r^{(i)} = s - \omega_i t$

end

3 + 1/(frequency) collectives

Combined two independent dot products into one global operation

Intermediate residual check is eliminated

Final residual checked infrequently
Improve Existing Algorithm

- Use successive over relaxation (SOR) for all variables except for gas pressure and solids volume fraction
  - SOR parallelizes better because it does not include any global collectives
  - SOR routine was updated to work with the latest version of the code
  - The gasifier test cases have been tested with this option and seem to run stably
  - Further improvements (e.g., red-black algorithm) to make SOR performance independent of processor decomposition are being considered
Develop New algorithm

- Evaluated two approaches for solving multiple equations in parallel
  - Functional decomposition
  - Interleaving communications and computations (in linear equation solver)
- Determined interleaving as the better method
- Evaluated a NBC library
Parallel Solving of Multiple Equations (New Algorithm)

Momentum ($l=0$)

Transport and physical properties, reaction rates ($l$)

Continuities ($l$)

Scalars ($l$)

Momentum ($l+1$)

Check convergence ($l$)

Solve these equations in parallel; e.g., over 20 equations for a gasifier model.

Parallelization approach

Wall clock time

Computational cost

Communication cost

Parallelization approach

$l$ – iteration number

$\text{Current}$ $\text{Interleaving}$ $\text{Functional decomposition}$
Improving I/O for Massively Parallel Environment

- **Problem:** Single files that accumulate all time records become unwieldy to handle and slow to read when the grid size is large
  - stopping the run and doing a restart requires user intervention and does not retain data from before the restart

- **Solution:** Create multiple files with a user defined time interval
  - Post-processing codes were modified to read this format
Improving I/O for Massively Parallel Environment

- Problem: Reading large files containing species mass fraction data was slow
- Solution:
  - Post-processing codes were modified to read data only for the requested species instead of all the species
  - Time required to post-process the results of a case on a 20 x 1176 x 40 grid, 16 species, and 100 time steps
    - Original code: 715 seconds
    - Modified code: 50 seconds
Actual simulated physical time per day for Case D with 10 million cells
Performance improvements achieved

High Resolution (10M grid) Gasifier Simulation on 2048 cores

- Phase 2 (Algorithm optimization & switched to Pathscale compiler from PGI)
- Phase 1 (PBS & MPI settings)
- Baseline

Simulated time (sec) in a day

New
Old

Baseline
Outline of Presentation

- MFIX-NG
- Polydispersed Flow
- High Resolution Gasifier
- Future Outlook
Goal

• Ensure that by 2015 multiphase science based computer simulations play a significant role in the design, operation, and troubleshooting of multiphase flow devices in fossil fuel processing plants.

• Benefits
  – Reduce the time and cost to develop efficient fossil fuel plants
  – Troubleshoot and mitigate problems
  – Enable the invention of novel reactor designs for next generation power plants and coal refineries
Collaboratory for Multiphase Flow Research (CMFR)

- Develop multiphase flow models and numerical techniques
- Validate the models with well calibrated experiments
- Promote the use of computational tools in industrial practice
- Provide a focal point for collaboration with academic and national labs
- Disseminate information and attract young researchers to the subject
Current Projects – CMFR

- Task 1: High Resolution Discretization Schemes for Multiphase Flow – TBD (WVU)
- Task 2: OpenFOAM Block Solver Development – Jasak (Wikki/WVU)
- Task 3: Evaluation and Benchmarking of Arches Code – Clarke (WVU)
- Task 4: Dispersion in CFB Riser: Effect of Riser Inlet Configuration – Johnson and Kang (WVU), Monazam (REM)
- Task 5: Discrete Particle Dynamics Simulations – McCarthy (Pitt), Higgs (CMU)
- Task 6: Coal Partitioning/Gasifier Fouling Project – Shadle (NETL), Kuhlman (WVU), Fruehan (CMU), Seetharaman (CMU), REM, NEA, PSU, LTI, SRI
- Task 7: Next generation multiphase flow solver – Prinkey (Aeolus/WVU)
- Task 8: Image Analysis of Circulating Fluid Bed Hydrodynamics – Ross (WVU)

Mineral particle dissolution in high temperature slag.
Extramural Projects

- **Dispersion Coefficient**
  - D. Gidaspow, IIT

- **Filtered two-fluid equations**
  - S. Sundaresan, Princeton U.

- **Kinetic theory of polydisperse systems**
  - C. Hrenya, U. Colorado
  - R. Fox, Iowa State U.
  - S. Subramaniam, Iowa State U.
  - S. Sundaresan, Princeton U.
  - R. Cocco, PSRI

- **Frictional flow Regime**
  - S. Sundaresan, Princeton U.
  - S. Subramaniam, Iowa State U.
  - G. Tardos, CCNY

Acknowledgements

• Dr. Sofiane Benyahia – DOE
• Dr. Aytekin Gel – Aeolus Research, Inc.
• Philip Nicoletti – Parsons, Inc.
• Dr. Thomas J. O’Brien – DOE
• Dr. Sreekanth Pannala – ORNL
• Dr. Badri Velamur Asokan – ORNL
• Michael Prinkey – Aeolus Research, Inc.
• Dr. Janine Galvin – DOE/U. Colorado
• Dr. Hrv Jasak – Wikki Ltd.
Visit Our Websites

Fossil Energy website:  www.fe.doe.gov

NETL website:  www.netl.doe.gov
References

- Iddir, H. and Arastoopour, H., 2005, AICHE J., 51, 6, pp. 1620-1632